A Construction of Polynomial Lattice Rules with Small Gain Coefficients

Jan Baldeaux*and Josef Dick†

Abstract

In this paper we construct polynomial lattice rules which have, in some sense, small gain coefficients using a component-by-component approach. The gain coefficients, as introduced by Owen, indicate to what degree the method improves upon Monte Carlo. We show that the variance of an estimator based on a scrambled polynomial lattice rule constructed component-by-component decays at a rate of $N^{-(2\alpha+1)+\delta}$, for all $\delta > 0$, assuming that the function under consideration has bounded variation of order α and where N denotes the number of quadrature points. An analogous result is obtained for Korobov polynomial lattice rules. It is also established that these rules are almost optimal for the function space considered in this paper. Furthermore, we discuss the implementation of the component-by-component approach and show how to reduce the computational cost associated with it. Finally, we present numerical results comparing scrambled polynomial lattice rules and scrambled digital nets.

Mathematics Subject Classification (2000): 65C05, 65D30, 65D32

1 Introduction

Quasi-Monte Carlo rules $\frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{x}_n)$, $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in [0,1]^s$, are equal weight integration formulas used to approximate integrals over the unit cube $[0,1]^s$, where s is typically

*The support of the Australian Research Council under its Centre of Excellence Program is gratefully acknowledged.

†The support of the Australian Research Council under its Centre of Excellence Program is gratefully acknowledged. The author is supported by an Australian Research Council Queen Elizabeth II Research Fellowship.

large. One can roughly divide quasi-Monte Carlo rules into lattice rules, see e.g. [18, 27], and digital nets, see e.g. [7, 18]. In this paper we focus on digital nets, the construction of which is based on linear algebra over finite fields, see [7, 18]. In particular, we are interested in a special case of digital nets, so-called polynomial lattice rules, which are constructed using polynomials over finite fields; polynomial lattice rules were introduced in [19], see also [5, 7, 18].

Studying the approximation of integrals using quasi-Monte Carlo methods, one wants to have information on the resulting integration errors. However, depending on the integrand under consideration, estimates of integration errors might be very conservative or unknown; a possible remedy to this problem is randomization, which allows us to obtain statistical information on integration errors, [23]. Popular choices of randomization methods are digital shifts, see e.g. [6, 7], and scrambling as introduced by Owen [23], see also [7, 10, 16, 24, 25, 30, 31, 32, 33]. In this paper, we focus on scrambling. In particular, we are interested in the variance of the estimator

$$\hat{I}(f) = \frac{1}{b^m} \sum_{h=0}^{b^m-1} f(\boldsymbol{y}_h), \tag{1}$$

where the points $\{\boldsymbol{y}_h\}_{h=0}^{b^m-1}$ are obtained by applying the scrambling algorithm to a polynomial lattice rule. Notice that $\hat{I}(f)$ is an unbiased estimator of $\int_{[0,1]^s} f(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$, that is, $\mathbb{E}(\hat{I}(f)) = \int_{[0,1]^s} f(\boldsymbol{x})\mathrm{d}\boldsymbol{x}$, see [23].

The variance of the estimator given in Equation (1) admits the representation, see [24],

$$\mathrm{Var}(\hat{I}(f)) = \frac{1}{N} \sum_{\boldsymbol{l} \in \mathbb{N}_0^s \setminus \{\boldsymbol{0}\}} \Gamma_{\boldsymbol{l}} \sigma_{\boldsymbol{l}}^2(f), \tag{2}$$

where N is the number of quadrature points. Equation (2) holds for any estimator obtained by applying the scrambling algorithm to a point set $\{\boldsymbol{x}_h\}_{h=0}^{b^m-1}$ such that $\boldsymbol{x}_h \in [0,1)^s$. Here, the values $\Gamma_{\boldsymbol{l}}$ are the so-called gain coefficients which depend only on the quadrature points and the values $\sigma_{\boldsymbol{l}}(f)$ depend only on the integrand f. They are derived from the crossed and nested Anova decomposition of f, see [24], and can be expressed in terms of Haar coefficients of the function f, see [24], or also as a sum of certain Walsh coefficients of f, see [7, Section 13.2]. In this sense, Equation (2) shows that $\mathrm{Var}(\hat{I}(f))$ can be expressed as a weighted sum of gain coefficients, where we interpret the $\sigma_{\boldsymbol{l}}^2(f)$ as weights.

In our investigations we consider a space of functions for which $\sigma_{\boldsymbol{l}}(f)$ has a certain

rate of decay. More precisely, for $0 < \alpha \leq 1$ we introduce a norm of the form

$$\|f\|_\alpha = \sup_{\boldsymbol{l} \in \mathbb{N}_0^s} b^{\alpha|\boldsymbol{l}|_1} \sigma_{\boldsymbol{l}}(f), \tag{3}$$

where $|\boldsymbol{l}|_1 = l_1 + \cdots + l_s$ for $\boldsymbol{l} = (l_1, \ldots, l_s)$. We show that α is related to the smoothness of f in the following sense:

If $f \in L_2([0,1]^s)$ has bounded variation of order α, then $\|f\|_\alpha < \infty$.

See Corollary 2.1 for details.

From (3) we obtain $\sigma_{\boldsymbol{l}}(f) \leq b^{-\alpha|\boldsymbol{l}|_1} \|f\|_\alpha$ and by substituting this formula into (2) we obtain

$$\mathrm{Var}(\hat{I}(f)) \leq \frac{1}{N} \sum_{\boldsymbol{l} \in \mathbb{N}_0^s \setminus \{\boldsymbol{0}\}} \Gamma_{\boldsymbol{l}} b^{-2\alpha|\boldsymbol{l}|_1} \|f\|_\alpha^2. \tag{4}$$

To construct polynomial lattice rules of high quality we use

$$\frac{1}{N} \sum_{\boldsymbol{l} \in \mathbb{N}_0^s \setminus \{\boldsymbol{0}\}} \Gamma_{\boldsymbol{l}} b^{-2\alpha|\boldsymbol{l}|_1} \tag{5}$$

as quality criterion. Notice that the sum (5) only depends on the quadrature points and not on the function f. We show that (5) has a simple closed form for any $0 < \alpha \leq 1$ which can easily be computed if the quadrature points are a digital net. The case $\alpha = 0$ needs to be excluded since in this case (5) is infinite.

Our aim is to find polynomial lattice rules for which the weighted sum of gain coefficients (5) is minimized. It is known from [24, 25, 31] that a small quality parameter t of a digital (t, m, s)-net yields small gain coefficients. In fact, one has $\Gamma_{\boldsymbol{l}} = 0$ for all $\boldsymbol{l} \in \mathbb{N}_0^s \setminus \{\boldsymbol{0}\}$ such that $|\boldsymbol{l}|_1 \leq m - t$. Here, on the other hand, we aim to minimize (5) since it can be used to bound the variance of the estimator $\mathrm{Var}(\hat{I}(f))$. In other words, we minimize the upper bound on the variance (4) for all functions f with $\|f\|_\alpha < \infty$ over the class of polynomial lattice rules.

We introduce additional parameters $\boldsymbol{\gamma} = (\gamma_j)_{1 \leq j \leq s}$ in the norm (3) in the sense of [29]. In this case the criterion (5) depends on the additional parameters $\boldsymbol{\gamma} = (\gamma_j)_{1 \leq j \leq s}$, in which case a small quality parameter t does not necessarily yield the smallest possible gain coefficients anymore. In such a situation component-by-component constructions [28] have proven useful since one can then optimize the quadrature points also with respect to the γ_j. This is also the approach taken here. More precisely, we show that by constructing polynomial lattice rules component-by-component one obtains a convergence of

$$\mathrm{Var}(\hat{I}(f)) = O(N^{-(2\alpha+1)+\delta}) \quad \text{for any } \delta > 0.$$

Apart from δ, which can be arbitrarily close to 0, this rate is best possible as shown in Section 6 for a large class of randomized algorithms. Further, if $\sum_{j=1}^{\infty} \gamma_j < \infty$, then the bound (4) does not depend on the dimension s. This result is stronger than what can be obtained for (t, m, s)-nets, since the increase of the t value of the form $t \approx s$ prevents one from obtaining a bound independent of the dimension only assuming that $\sum_{j=1}^{\infty} \gamma_j < \infty$. This condition is also necessary, see [32] for a result on a related space. Hence the rules we construct here are simultaneously optimal in terms of the convergence rate as well as in terms of their dependence on the dimension.

Notice that the additional parameters $\boldsymbol{\gamma} = (\gamma_j)_{j \geq 1}$, as introduced in [29], have been found to be very useful from both a theoretical and a practical point of view. They allow us to associate more importance with some variables than with others, which ties in nicely with concepts such as effective dimension, see, e.g., [2]. This has been used to explain the success of quasi-Monte Carlo rules in finance.

We now give the structure of the paper. In Section 2, we recall the definition of polynomial lattice rules, the scrambling algorithm and define the function space studied in this paper. The variance of estimators based on scrambled polynomial lattice rules is studied in Section 3. Next, in Sections 4 and 5, we construct polynomial lattice rules for which the variance of the associated estimator converges at a rate of $N^{-(1+2\alpha)+\delta}$, for all $\delta > 0$. The constructions are based on a component-by-component approach and the Korobov construction respectively. In Section 6, we define a large class of randomized algorithms, which includes adaptive ones, and consequently establish that the variance of any estimator based on an algorithm from this class converges at most a rate of $N^{-(1+2\alpha)}$ for the function space under consideration in this paper. Hence our constructions are almost optimal for the class of algorithms defined in Section 6. In Section 7, we study the implementation of the component-by-component approach, in particular, we show how to reduce the computational effort associated with it. This implementation is made use of in Section 8, where we compare the performance of scrambled polynomial lattice rules constructed in Section 4 to the performance of scrambled digital nets.

2 Preliminaries

In this section, we define polynomial lattice rules, recall the scrambling algorithm and introduce the function space under consideration in this paper.

2.1 Polynomial Lattice Rules

Polynomial lattice rules were introduced in [19], see also [5, 7, 18]. We fix a prime b and denote by $\mathbb{Z}_b$ the finite field containing b elements and by $\mathbb{Z}_b((x^{-1}))$ the field of formal Laurent series over $\mathbb{Z}_b$. Elements of $\mathbb{Z}_b((x^{-1}))$ are formal Laurent series,

$$L = \sum_{l=w}^{\infty} t_l x^{-l}\,,$$

where w is an arbitrary integer and all $t_l \in \mathbb{Z}_b$. The field $\mathbb{Z}_b((x^{-1}))$ contains the field of rational functions over $\mathbb{Z}_b$ as a subfield. Finally, the set of polynomials over $\mathbb{Z}_b$ is denoted by $\mathbb{Z}_b[x]$. For an integer m, we denote by v_m the map from $\mathbb{Z}_b((x^{-1}))$ to $[0,1)$ defined by

$$v_m\left(\sum_{l=w}^{\infty} t_l x^{-l}\right) = \sum_{l=\max(1,w)}^{m} t_l b^{-l}\,.$$

The following definition of polynomial lattice rules stems from [19], see also [7, 18].

Definition 2.1 *Let b be prime and m be an integer. For a given dimension $s \geq 1$, choose $p(x) \in \mathbb{Z}_b[x]$ with $deg(p(x)) = m$ and $q_1(x), \ldots, q_s(x) \in \mathbb{Z}_b[x]$. For $0 \leq h < b^m$ let $h = h_0 + h_1 b + \cdots + h_{m-1} b^{m-1}$ be the b-adic expansion of h. With each such h we associate the polynomial*

$$\overline{h}(x) = \sum_{r=0}^{m-1} h_r x^r \in \mathbb{Z}_b[x]\,.$$

Then $S_{p,m}(\boldsymbol{q})$, where $\boldsymbol{q} = (q_1, \ldots, q_s)$, is the point set consisting of the b^m points

$$\boldsymbol{x}_h = \left(v_m\left(\frac{\overline{h}(x)q_1(x)}{p(x)}\right), \ldots, v_m\left(\frac{\overline{h}(x)q_s(x)}{p(x)}\right)\right) \in [0,1)^s\,,$$

for $0 \leq h < b^m$. A quasi-Monte Carlo rule using the point set $S_{p,m}(\boldsymbol{q})$ is called a polynomial lattice rule.

We remark that polynomial lattice point sets are also digital nets, see [7, 17, 18].

For the remainder of the paper, we use the following notation: We write $\vec{h}$ for vectors over $\mathbb{Z}_b$ and $\boldsymbol{h}$ for vectors over $\mathbb{Z}$ or $\mathbb{R}$. Polynomials over $\mathbb{Z}_b$ are denoted by $h(x)$ and vectors of polynomials by $\boldsymbol{h}(x)$. Furthermore, given an integer h with b-adic expansion $h = \sum_{r=0}^{\infty} h_r b^r$, we denote the associated polynomial by

$$\overline{h}(x) = \sum_{r=0}^{\infty} h_r x^r\,.$$

For arbitrary $\boldsymbol{h}(x) = (h_1(x), \ldots, h_s(x)) \in \mathbb{Z}_b[x]^s$ and $\boldsymbol{q}(x) = (q_1(x), \ldots, q_s(x)) \in \mathbb{Z}_b[x]^s$, we define the "inner product"

$$\boldsymbol{h}(x) \cdot \boldsymbol{q}(x) = \sum_{j=1}^{s} h_j q_j(x) \in \mathbb{Z}_b[x]$$

and we write $q(x) \equiv 0 \pmod{p(x)}$ if $p(x)$ divides $q(x)$ in $\mathbb{Z}_b[x]$.

Finally, we introduce the dual lattice which plays an important role in numerical integration, see [5, 7], which requires us to introduce the following function: For a non-negative integer k with b-adic expansion $k = k_0 + k_1 b + \ldots$ we write $\mathrm{tr}_m(k) = k_0 + k_1 b + \cdots + k_{m-1} b^{m-1}$ and thus the associated polynomial

$$\mathrm{tr}_m(k)(x) = k_0 + k_1 x + \ldots k_{m-1} x^{m-1} \in \mathbb{Z}_b[x]$$

has degree $< m$. For a vector $\boldsymbol{k} \in \mathbb{N}_0^s$, $\mathrm{tr}_m(\boldsymbol{k})$ is defined componentwise.

Definition 2.2 *Let $\boldsymbol{q}(x) = (q_1(x), \ldots, q_s(x)) \in \mathbb{Z}_b[x]^s$, then the dual polynomial lattice of $S_{p,m}(\boldsymbol{q})$ is given by*

$$\begin{aligned} \mathscr{D} &= \mathscr{D}_p(\boldsymbol{q}) \\ &= \{\boldsymbol{k} \in \mathbb{N}_0^s : \\ &\qquad \mathrm{tr}_m(k_1)(x) + \cdots + \mathrm{tr}_m(k_2)(x) q_2 + \cdots + \mathrm{tr}_m(k_s)(x) q_s(x) \equiv \boldsymbol{0} \pmod{p(x)}\}. \end{aligned}$$

2.2 The Scrambling Algorithm

The scrambling algorithm is a randomization algorithm which was introduced by Owen, see [23] and also [8, 10, 24, 25, 30, 31, 33].

We now describe the scrambling algorithm using a generic point $\boldsymbol{x} \in [0,1)^s$, where $\boldsymbol{x} = (x_1, \ldots, x_s)$ and

$$x_j = \frac{\xi_{j,1}}{b} + \frac{\xi_{j,2}}{b^2} + \ldots.$$

Then the scrambled point shall be denoted by $\boldsymbol{y} \in [0,1)^s$, where $\boldsymbol{y} = (y_1, \ldots, y_s)$,

$$y_j = \frac{\eta_{j,1}}{b} + \frac{\eta_{j,2}}{b^2} + \ldots.$$

The permutation applied to $\xi_{j,l}$, $j = 1, \ldots, s$ depends on $\xi_{j,k}$, for $1 \le k < l$. In particular, $\eta_{j,1} = \pi_j(\xi_{j,1})$, $\eta_{j,2} = \pi_{j,\xi_{j,1}}(\xi_{j,2})$, $\eta_{j,3} = \pi_{j,\xi_{j,1},\xi_{j,2}}(\xi_{j,3})$ and in general

$$\eta_{j,k} = \pi_{j,\xi_{j,1},\ldots,\xi_{j,k-1}}(\xi_{j,k}), k \ge 2,$$

where π_j and $\pi_{j,\xi_{j,1},\ldots,\xi_{j,k-1}}$, $k \geq 2$ are random permutations of $\{0,1,\ldots,b-1\}$. We assume that permutations with different indices are mutually independent. Also, if we apply the scrambling algorithm to $\boldsymbol{x}$ to obtain $\boldsymbol{y}$, then $\boldsymbol{y}$ is uniformly distributed in $[0,1)^s$, see [23, Proposition 2]. Finally, it was shown in [23] that the scrambling algorithm preserves the (t,m,s)-net property with probability 1, i.e applying the scrambling algorithm to the points of a (t,m,s)-net results in a (t,m,s)-net with probability 1.

2.3 A Weighted Walsh Function Space based on Variance

In this section, we introduce the function space under consideration in this paper. In particular, we consider weighted spaces and for this purpose, we introduce a sequence of positive, non-increasing weights $\boldsymbol{\gamma} = (\gamma_j)_{j=1}^{\infty}$. The purpose of the weights is to model the importance of the different variables and we point out that the idea stems from [29]. For $s \in \mathbb{N}$, let $[s] = \{1,\ldots,s\}$ and for $\mathfrak{u} \subseteq [s]$ let $\boldsymbol{\gamma}_{\mathfrak{u}} := \prod_{j\in\mathfrak{u}} \gamma_j$ be the weight associated with the projection onto coordinates whose index is contained in $\mathfrak{u}$.

Walsh functions have been an important tool in the analysis of digital nets; in [14], Walsh functions were used for the first time to analyze nets and the connection between numerical integration using digital nets and Walsh functions was made in [6], see also [7].

We now briefly recall the definition of Walsh functions. Let $\mathbb{N}_0$ denote the set of nonnegative and $\mathbb{N}$ the set of positive integers. Each $k \in \mathbb{N}$ has a unique b-adic representation $k = \sum_{i=0}^{a} k_i b^i$ with digits $k_i \in \{0,\ldots,b-1\}$ for $0 \leq i \leq a$, where $k_a \neq 0$. For $k = 0$ we have $a = 0$ and $k_0 = 0$. Similarly, each $x \in [0,1)$ has a b-adic representation $x = \sum_{i=1}^{\infty} \xi_i b^{-i}$ with digits $\xi_i \in \{0,\ldots,b-1\}$ for $i \geq 1$. This representation is unique in the sense that infinitely many of the ξ_i must differ from $b-1$. We define the kth Walsh function in base b, $\mathrm{wal}_k : [0,1) \to \mathbb{C}$ by

$$\mathrm{wal}_k(x) := \exp(2\pi\mathtt{i}(\xi_1 k_0 + \cdots + \xi_{a+1}k_a)/b)\,.$$

For dimension $s \geq 2$ and vectors $\boldsymbol{k} = (k_1,\ldots,k_s) \in \mathbb{N}_0^s$ and $\boldsymbol{x} = (x_1,\ldots,x_s) \in [0,1)^s$ we define $\mathrm{wal}_{\boldsymbol{k}} : [0,1)^s \to \mathbb{C}$ by

$$\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x}) := \prod_{j=1}^{s} \mathrm{wal}_{k_j}(x_j)\,.$$

Studying integration errors resulting from the approximation of an integral based on a digital net or a polynomial lattice rule, it is useful to consider the Walsh series of the integrand f. For $f \in L_2([0,1]^s)$, the Walsh series of f is given by

$$f(\boldsymbol{x}) \sim \sum_{\boldsymbol{k}\in\mathbb{N}_0^s} \hat{f}(\boldsymbol{k})\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x})\,, \tag{6}$$

where the Walsh coefficients $\hat{f}(\boldsymbol{k})$ are given by

$$\hat{f}(\boldsymbol{x}) = \int_{[0,1]^s} f(\boldsymbol{x})\overline{\mathrm{wal}_{\boldsymbol{k}}(\boldsymbol{x})}d\boldsymbol{x}\,.$$

We do not necessarily have equality in Equation (6), however, the completeness of the Walsh function system $\{\mathrm{wal}_{\boldsymbol{k}} : \boldsymbol{k} \in \mathbb{N}_0^s\}$ (see for instance [7, Appendix A]), implies that

$$\mathrm{Var}[f] = \sum_{\boldsymbol{k}\in\mathbb{N}_0^s\setminus\{\boldsymbol{0}\}} |\hat{f}(\boldsymbol{k})|^2\,,$$

where $\mathrm{Var}[f] = \int_{[0,1]^s}(f(\boldsymbol{x}) - \overline{f})^2 d\boldsymbol{x}$, and where $\overline{f} = \int_{[0,1]^s} f(\boldsymbol{x})d\boldsymbol{x}$.

Let $\sigma^2_{(\boldsymbol{l}_{\mathfrak{u}},\boldsymbol{0})} = \sum_{\boldsymbol{k}\in L_{(\boldsymbol{l}_{\mathfrak{u}},\boldsymbol{0})}} |\hat{f}(\boldsymbol{k})|^2$, where

$$L_{(\boldsymbol{l}_{\mathfrak{u}},\boldsymbol{0})} = \left\{\boldsymbol{k} \in \mathbb{N}_0^s : b^{l_i-1} \le k_i < b^{l_i} \text{ for } i \in \mathfrak{u} \text{ and } k_i = 0 \text{ for } i \in [s]\setminus\mathfrak{u}\right\}. \tag{7}$$

Further let $|\boldsymbol{l}|_1 = \sum_{j=1}^s l_j$ for $\boldsymbol{l} = (l_1,\ldots,l_s)$. For $0 < \alpha \le 1$ we define a weighted norm for functions $f \in L_2([0,1]^s)$ by

$$\|f\|_\alpha = \max_{\mathfrak{u}\subseteq[s]} \gamma_{\mathfrak{u}}^{-1/2} \sup_{\boldsymbol{l}_{\mathfrak{u}}\in\mathbb{N}^{|\mathfrak{u}|}} b^{\alpha|\boldsymbol{l}_{\mathfrak{u}}|_1}\sigma_{(\boldsymbol{l}_{\mathfrak{u}},\boldsymbol{0})}(f). \tag{8}$$

For $0 < \alpha \le 1$ define a space $V_{\alpha,s,\boldsymbol{\gamma}} \subseteq L_2([0,1]^s)$ consisting of all functions f for which $\|f\|_\alpha < \infty$. (One could of course use some ℓ_p norm instead of the supremum-norm to define $\|\cdot\|_\alpha$ and the function space, but these do not yield a quality criterion of the form (5) which can be used for our construction, see (11) and Lemma 3.1 below.)

The following observation stems from [7, Section 13.5]: For a subinterval $J = \prod_{i=1}^s [x_i, y_i)$ with $0 \le x_i < y_i \le 1$ and a function $f : [0,1)^s \to \mathbb{R}$, let the function $\Delta(f, J)$ denote the alternating sum of f at the vertices of J where adjacent vertices have opposite signs. (Hence for $f = \prod_{i=1}^s f_i$ we have $\Delta(f, J) = \prod_{i=1}^s (f_i(x_i) - f_i(y_i))$.)

We define the generalized variation in the sense of Vitali of order $0 < \alpha \le 1$ by

$$V_\alpha^{(s)}(f) = \sup_{\mathcal{P}} \left(\sum_{J\in\mathcal{P}} \mathrm{Vol}(J) \left|\frac{\Delta(f,J)}{\mathrm{Vol}(J)^\alpha}\right|^2\right)^{1/2},$$

where the supremum is extended over all partitions $\mathcal{P}$ of $[0,1]^s$ into subintervals and $\mathrm{Vol}(J)$ denotes the volume of the subinterval J.

For $\alpha = 1$ and if the partial derivatives of f are continuous on $[0,1]^s$ we also have the formula

$$V_1^{(s)}(f) = \left(\int_{[0,1]^s} \left|\frac{\partial^s f}{\partial x_1 \cdots \partial x_s}\right|^2 d\boldsymbol{x}\right)^{1/2}.$$

Until now we did not take projections to lower-dimensional faces into account.

For $\emptyset \neq u \subseteq [s]$, let $V_\alpha^{(|u|)}(f_u;u)$ be the generalized Vitali variation with coefficient $0 < \alpha \leq 1$ of the $|u|$-dimensional function

$$f_u(\boldsymbol{x}_u) = \int_{[0,1)^{s-|u|}} f(\boldsymbol{x}) d\boldsymbol{x}_{[s]\setminus u}.$$

For $u = \emptyset$ we have $f_\emptyset = \int_{[0,1)^s} f(\boldsymbol{x}) d\boldsymbol{x}_{[s]}$ and we define $V_\alpha^{(|\emptyset|)}(f_\emptyset;\emptyset) = |f_\emptyset|$. Then

$$V_\alpha(f) = \left(\sum_{u \subseteq [s]} \left(V_\alpha^{(|u|)}(f_u;u) \right)^2 \right)^{1/2} \tag{9}$$

is called the generalized Hardy and Krause variation of f on $[0,1]^s$.

A function f for which $V_\alpha(f) < \infty$ is said to be of finite variation of order α.

The following result is from [4] and [7, Section 13.5].

Corollary 2.1 *Let $b \geq 2$ be a natural number and let $f \in L_2([0,1]^s)$ have bounded variation $V_\alpha(f) < \infty$ of order $0 < \alpha \leq 1$. Then*

$$\|f\|_\alpha \leq \max\left(\|f\|_{L_2} \gamma_\emptyset^{-1}, V_\alpha(f) \max_{\emptyset \neq \mathfrak{u} \subseteq [s]} \gamma_{\mathfrak{u}}^{-1/2} (b-1)^{(\alpha-1/2)_+ |\mathfrak{u}|} \right).$$

Hence every function $f \in L_2([0,1]^s)$ which has bounded variation of order $0 < \alpha \leq 1$ is in $V_{\alpha,s,\boldsymbol{\gamma}}$. For the extreme case $\alpha = 0$ one obtains $V_{0,s,\boldsymbol{\gamma}} = L_2([0,1]^s)$, but this case is not included in our investigations since the criterion (5) is infinite in this case, see (11) and Lemma 3.1 below.

3 The Variance of Estimators based on Scrambled Polynomial Lattice Rules

In this section, we discuss the variance of the estimator

$$\hat{I}(f) = \frac{1}{b^m} \sum_{h=0}^{b^m-1} f(\boldsymbol{y}_h), \tag{10}$$

where the points $\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{b^m-1}$ are obtained by applying the scrambling algorithm to a digital (t,m,s)-net over $\mathbb{Z}_b$.

We use the following notation: For a non-negative integer k with b-adic expansion

$$k = k_0 + k_1 b + \dots,$$

we write $\vec{k} = (k_0, k_1, \dots)^\top$, which is an infinite-dimensional vector, and we use

$$\mathrm{tr}_m(\vec{k}) = (k_0, k_1, \dots, k_{m-1})^\top.$$

We now introduce the integration problem studied in this paper, in particular, we are interested in the worst-case variance of multivariate integration in $V_{\alpha,s,\boldsymbol{\gamma}}$ using a scrambled quasi-Monte Carlo rule $Q_{b^m,s}$:

$$\mathrm{Var}(Q_{b^m,s}, V_{\alpha,s,\boldsymbol{\gamma}}) = \sup_{f \in V_{\alpha,s,\boldsymbol{\gamma}}, \|f\|_\alpha \le 1} \mathrm{Var}[\hat{I}(f, Q_{b^m,s})],$$

where $\hat{I}(f, Q_{b^m,s})$ denotes the estimator based on the point set obtained by applying the scrambling algorithm to $Q_{b^m,s}$. We denote the quasi-Monte Carlo rule based on a polynomial lattice rule $S_{p,m}(\boldsymbol{q})$ by $Q_{b^m,s}(\boldsymbol{q})$ and the associated worst-case variance by $\mathrm{Var}(Q_{b^m,s}(\boldsymbol{q}), V_{\alpha,s,\boldsymbol{\gamma}})$. For $k = \kappa_0 + \kappa_1 b + \cdots + \kappa_{a-1} b^{a-1} \in \mathbb{N}_0$ let

$$r_{\alpha,\gamma}(k) = \begin{cases} 1 & \text{if } k = 0, \\ \gamma \frac{b}{(b-1)b^{\alpha a}} & \text{if } k > 0. \end{cases}$$

For $\boldsymbol{k} = (k_1, \dots, k_s) \in \mathbb{N}_0^s$ let $r_{\alpha,\boldsymbol{\gamma}}(\boldsymbol{k}) = \prod_{j=1}^s r_{\alpha,\gamma_j}(k_j)$.

The next corollary gives a bound on the quantity $\mathrm{Var}(Q_{b^m,s}(\boldsymbol{q}), V_{\alpha,s,\boldsymbol{\gamma}})$.

Corollary 3.1 *Let $0 < \alpha \le 1$, $\boldsymbol{q} \in \mathbb{Z}_b[x]^s$ be a generating vector for a polynomial lattice rule with modulus p, and $\mathrm{Var}(Q_{b^m,s}(\boldsymbol{q}), V_{\alpha,s,\boldsymbol{\gamma}})$ be defined as above. Then*

$$\mathrm{Var}(Q_{b^m,s}(\boldsymbol{q}), V_{\alpha,s,\boldsymbol{\gamma}}) \le \sum_{\boldsymbol{k} \in \mathscr{D}_p(\boldsymbol{q}) \setminus \{\boldsymbol{0}\}} r_{2\alpha+1,\boldsymbol{\gamma}}(\boldsymbol{k}),$$

where $\mathscr{D}_p(\boldsymbol{q})$ is the dual polynomial lattice.

Proof. The corollary follows from the following facts: For any $f \in L_2([0,1]^s)$, let $\hat{I}(f)$ be given by Equation (10) and $\{\boldsymbol{x}_0, \dots, \boldsymbol{x}_{b^m-1}\}$ be a digital (t,m,s)-net over $\mathbb{Z}_b$ with generating matrices $C_1, \dots, C_s$ over $\mathbb{Z}_b$, then we have (see [31] or also [7, Section 13.5])

$$\mathrm{Var}[\hat{I}(f)] = \sum_{\emptyset \ne \mathfrak{u} \subseteq [s]} \frac{b^{|\mathfrak{u}|}}{(b-1)^{|\mathfrak{u}|}} \sum_{\boldsymbol{l}_\mathfrak{u} \in \mathbb{N}^{|\mathfrak{u}|}} \frac{\sigma^2_{(\boldsymbol{l}_\mathfrak{u}, \boldsymbol{0})}(f)}{b^{|\boldsymbol{l}_\mathfrak{u}|_1}} |L_{(\boldsymbol{l}_\mathfrak{u}, \boldsymbol{0})} \cap \mathscr{D}(C_1, \dots, C_s)|,$$

where $\mathscr{D}(C_1,\dots,C_s) = \left\{\boldsymbol{k} \in \mathbb{N}_0^s : C_1^\top \mathrm{tr}_m(\vec{k}_1) + \cdots + C_s^\top \mathrm{tr}_m(\vec{k}_s) = \vec{0}\right\}$ and $L_{(\boldsymbol{l}_\mathfrak{u},\boldsymbol{0})}$ is given in Equation (7). Furthermore, if the $C_1,\dots,C_s$ are the generating matrices of the point set $S_{p,m}(\boldsymbol{q})$, then for any $\boldsymbol{k} \in \mathbb{N}_0^s \setminus \{\boldsymbol{0}\}$ we have

$$C_1^\top \mathrm{tr}_m(\vec{k}_1) + \cdots + C_s^\top \mathrm{tr}_m(\vec{k}_s) = \vec{0} \Leftrightarrow \mathrm{tr}_m(\boldsymbol{k}) \cdot \boldsymbol{q} \equiv 0 \pmod{p},$$

which was first established in [18, Lemma 4.40]. Using (8) and

$$\sum_{\emptyset \neq \mathfrak{u} \subseteq [s]} \frac{b^{|\mathfrak{u}|}}{(b-1)^{|\mathfrak{u}|}} \sum_{\boldsymbol{l}_\mathfrak{u} \in \mathbb{N}^{|\mathfrak{u}|}} \frac{1}{b^{(2\alpha+1)|\boldsymbol{l}_\mathfrak{u}|_1}} |L_{(\boldsymbol{l}_\mathfrak{u},\boldsymbol{0})} \cap \mathscr{D}_p(\boldsymbol{q})| = \sum_{\boldsymbol{k} \in \mathscr{D}_p(\boldsymbol{q}) \setminus \{\boldsymbol{0}\}} r_{2\alpha+1,\boldsymbol{\gamma}}(\boldsymbol{k}),$$

the result follows. □

We denote the bound in Corollary 3.1 by

$$B(\boldsymbol{q},\alpha,\boldsymbol{\gamma}) := \sum_{\boldsymbol{k} \in \mathscr{D}_p(\boldsymbol{q}) \setminus \{\boldsymbol{0}\}} r_{2\alpha+1,\boldsymbol{\gamma}}(\boldsymbol{k}). \tag{11}$$

This bound is almost the same as the square worst case error for integration in a certain Walsh space considered in [5], see in particular [5, Lemma 4.1].

As in [5], $B(\boldsymbol{q},\alpha,\boldsymbol{\gamma})$ can easily be computed and therefore be used as a quality criterion for polynomial lattice rules. We write $\log_b$ for the logarithm in base b and we set $b^{2\alpha\lfloor \log_b 0 \rfloor} = 0$.

Lemma 3.1 *Let $B(\boldsymbol{q},\alpha,\boldsymbol{\gamma})$ be given by Equation* (11)*. Then*

$$B(\boldsymbol{q},\alpha,\boldsymbol{\gamma}) = -1 + \frac{1}{b^m} \sum_{h=0}^{b^m-1} \prod_{j=1}^{s} \left(1 + \frac{b}{b-1} \gamma_j \phi_\alpha(x_{h,j})\right), \tag{12}$$

where for $x \in [0,1)$ we set

$$\phi_\alpha(x) = \frac{b - 1 - b^{2\alpha\lfloor \log_b x \rfloor}(b^{2\alpha+1}-1)}{b(b^{2\alpha}-1)}.$$

A detailed proof of this result can be found in [1].

In the next remark, we show that if we construct a polynomial lattice rule which achieves optimal convergence rates for functions in $V_{\alpha,s,\boldsymbol{\gamma}}$ for some given $0 < \alpha \leq 1$, then this polynomial lattice rule also achieves optimal convergence rates for functions in $V_{\alpha',s,\boldsymbol{\gamma}'}$ where $\alpha \leq \alpha' \leq 1$. This means that the polynomial lattice rule constructed to achieve optimal convergence rates for functions of smoothness α adjusts itself to the optimal rate of convergence, as long as the smoothness α' of the function under consideration satisfies $\alpha' \geq \alpha$.

Remark 3.1 *Assume that for a fixed α, $0 < \alpha \leq 1$, we have constructed a polynomial lattice rule $S_{p,m}(\boldsymbol{q})$ such that*

$$B(\boldsymbol{q}, \alpha, \boldsymbol{\gamma}) \leq C_{s,\alpha,\boldsymbol{\gamma},\delta} N^{-(1+2\alpha)+\delta}, \tag{13}$$

for all $\delta > 0$, where $C_{s,\alpha,\boldsymbol{\gamma},\delta}$ is permitted to depend on $s, \alpha, \boldsymbol{\gamma}$ and δ. We point out that explicit constructions of polynomial lattice rules satisfying Equation (13) *are given in Sections 4 and 5. It follows immediately from Jensen's inequality, that*

$$B(\boldsymbol{q}, \alpha, \boldsymbol{\gamma})^{\frac{1+2\alpha'}{1+2\alpha}} \geq B(\boldsymbol{q}, \alpha', \boldsymbol{\gamma}^{\frac{1+2\alpha'}{1+2\alpha}}),$$

for $\alpha \leq \alpha' \leq 1$. Making use of Assumption (13)*, we conclude that*

$$B(\boldsymbol{q}, \alpha', \boldsymbol{\gamma}^{\frac{1+2\alpha'}{1+2\alpha}}) \leq C_{s,\alpha,\boldsymbol{\gamma},\delta}^{\frac{1+2\alpha'}{1+2\alpha}} \, N^{-(1+2\alpha')+\delta\frac{1+2\alpha'}{1+2\alpha}},$$

for all $\delta > 0$. In particular, this observation motivates the construction of polynomial lattice rules for which $\alpha < 1$, as the resulting point sets still achieve optimal convergence rates for functions of bounded variation of order α', where $\alpha \leq \alpha' \leq 1$.

4 Component-By-Component Construction of Polynomial Lattice Rules

In this section, we show how to construct a polynomial lattice rule using a component-by-component approach so that the bound given in Equation (11) converges at a rate of $N^{-1-2\alpha+\delta}$, for any $\delta > 0$. We remark that in [7, Theorem 13.24], the corresponding result for digital nets was presented. A component-by-component (CBC) approach was first considered in [28] in the context of constructing lattice rules. Subsequently, the CBC algorithm has been applied to the construction of polynomial lattice rules in [5].

We use $R_{b,m}$ to denote the set of all non-zero polynomials in $\mathbb{Z}_b[x]$ with degree at most $m-1$, i.e.

$$R_{b,m} := \{q \in \mathbb{Z}_b[x] : deg(q) < m \text{ and } q \neq 0\} .$$

It is clear that $|R_{b,m}| = b^m - 1$ and furthermore it follows from the construction principle described in Subsection 2.1 that the polynomials q_j can be restricted to $R_{b,m}$. Algorithm 1 gives the CBC algorithm.

The next theorem shows that Algorithm 1 indeed constructs a $\boldsymbol{q}_d^* \in R_{b,m}^d$ so that $B((q_1^*, \ldots, q_d^*), \alpha, \boldsymbol{\gamma})$ converges at a rate of $N^{-1-2\alpha+\delta}$, for any $\delta > 0$.

Algorithm 1 CBC algorithm

Require: b a prime, $s, m \in \mathbb{N}$ and weights $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$.

1: Choose an irreducible polynomial $p \in \mathbb{Z}_b[x]$, with $\deg(p) = m$.
2: Set $q_1 = 1$.
3: **for** $d = 2$ to s **do**
4: find $q_d \in R_{b,m}$ by minimizing $B((q_1, \ldots, q_d), \alpha, \boldsymbol{\gamma})$ as a function of q_d.
5: **end for**
6: **return** $\boldsymbol{q} = (q_1, \ldots, q_s)$.

Theorem 4.1 *Let b be prime and $p \in \mathbb{Z}_b[x]$ be irreducible, with $deg(p) = m \geq 1$. Suppose $(q_1^*, \ldots, q_s^*) \in R_{b,m}^s$ is constructed using Algorithm 1. Then for all $d = 1, \ldots, s$ we have*

$$B((q_1^*, \ldots, q_d^*), \alpha, \boldsymbol{\gamma}) \leq \frac{1}{(b^m - 1)^{1/\lambda}} \prod_{j=1}^{d} \left[1 + \gamma_j^\lambda C_{b,\alpha,\lambda}\right]^{1/\lambda},$$

for all $\frac{1}{2\alpha+1} < \lambda \leq 1$ where

$$C_{b,\alpha,\lambda} = \max\left(\frac{1}{(b^{2\alpha} - 1)^\lambda}, \frac{(b-1)^{1-\lambda}}{b^{2\alpha\lambda} - b^{1-\lambda}}\right). \tag{14}$$

A proof of this result can be obtained by making a few modifications to the proof of [5, Theorem 4.4], which is included in [1]. The additional term $(b^{2\alpha} - 1)^{-\lambda}$ in the definition of $C_{b,\alpha,\lambda}$ arises from the one-dimensional case for which we have (we assume $k = \kappa_0 + \kappa_1 b + \cdots + \kappa_{a-1} b^{a-1}$)

$$\begin{aligned}
B((1), \alpha, \boldsymbol{\gamma}) &= \gamma_1 \frac{b}{b-1} \sum_{k=1, b^m | k}^{\infty} b^{-\alpha a} \\
&= \gamma_1 \frac{b}{b-1} \sum_{l=m+1}^{\infty} (b-1) b^{l-m-1} b^{-(2\alpha+1)l} \\
&= \frac{1}{b^{(2\alpha+1)m}} \frac{\gamma_1}{b^{2\alpha} - 1} \\
&\leq \frac{1}{(b^m - 1)^{1/\lambda}} [1 + \gamma_1^\lambda (b^{2\alpha} - 1)^{-\lambda}]^{1/\lambda},
\end{aligned}$$

for all $\frac{1}{2\alpha+1} < \lambda \leq 1$. The induction with respect to the dimension can be carried out as in the proof of [5, Theorem 4.4].

The next result discusses the tractability of Algorithm 1.

Corollary 4.1 *Let b be prime, $p \in \mathbb{Z}_b[x]$ be irreducible with $deg(p) = m \geq 1$ and $N = b^m$. Suppose $\boldsymbol{q}_s^* \in R_{b,m}^s$ is constructed using Algorithm 1. Then we have the following:*

1. $$B(\boldsymbol{q}_s^*, \alpha, \boldsymbol{\gamma}) \leq c_{s,\alpha,\boldsymbol{\gamma},\delta}\ (N-1)^{-(2\alpha+1)+\delta}\ ,\ \textit{for all } 0 < \delta \leq 2\alpha,$$

   where

   $$c_{s,\alpha,\boldsymbol{\gamma},\delta} = \prod_{j=1}^{s} \left[1 + \gamma_j^{\frac{1}{2\alpha+1-\delta}} C_{b,\alpha,(2\alpha+1-\delta)^{-1}}\right]^{2\alpha+1-\delta}.$$

2. *Assume*

   $$\sum_{j=1}^{\infty} \gamma_j^{\frac{1}{2\alpha+1-\delta}} < \infty. \tag{15}$$

   Then $c_{s,\alpha,\boldsymbol{\gamma},\delta} \leq c_{\infty,\alpha,\boldsymbol{\gamma},\delta} < \infty$ and we have

   $$B(\boldsymbol{q}_s^*, \alpha, \boldsymbol{\gamma}) \leq c_{\infty,\alpha,\boldsymbol{\gamma},\delta}(N-1)^{-(2\alpha+1)+\delta}\ ,\ \textit{for all } 0 < \delta \leq 2\alpha.$$

   Thus the bound $B(\boldsymbol{q}_s^*, \alpha, \boldsymbol{\gamma})$ is bounded independently of the dimension.

3. *Under the assumption*

   $$A := \limsup_{s\to\infty} \frac{\sum_{j=1}^{s} \gamma_j}{s} < \infty,$$

   we obtain $c_{s,\alpha,\boldsymbol{\gamma},2\alpha} \leq \tilde{c}_\eta (b-1)^{2\alpha} s^{\frac{A+\eta}{b^{2\alpha}-1}}$ and therefore

   $$B(\boldsymbol{q}_s^*, \alpha, \boldsymbol{\gamma}) \leq \tilde{c}_\eta s^{\frac{A+\eta}{b^{2\alpha}-1}} (N-1)^{-1}$$

   for all $\eta > 0$, where the constant $\tilde{c}_\eta$ only depends on η. Thus the bound $B(\boldsymbol{q}_s^*, \alpha, \boldsymbol{\gamma})$ satisfies a bound which depends only polynomially on the dimension.

The proof is similar to the proof of [5, Corollary 4.5] and can be found in [1].

5 Construction of Korobov Polynomial Lattice Rules

In this section, we construct Korobov polynomial lattice rules. The ideas underlying this algorithm stem from the construction of lattice rules, see [12]. We remark that the construction of Korobov polynomial lattice rules has been examined in [5], see also [13]. We denote the generating vector for the Korobov polynomial lattice rule by $\psi(q) =$

Algorithm 2 Korobov algorithm

Require: b a prime, $s, m \in \mathbb{N}$ and weights $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$.

1: Choose an irreducible polynomial $p \in \mathbb{Z}_b[x]$, with $\deg(p) = m$.

2: Find $q^* \in R_{b,m}$ by minimizing $B(\psi(q), \alpha, \boldsymbol{\gamma})$.

$(1, q, \ldots, q^{s-1}) \pmod p$. As in Section 4, we work with the bound $B(\psi(q), \alpha, \boldsymbol{\gamma})$ and now state the algorithm showing how to construct Korobov polynomial lattice rules.

We obtain the following bound for $B(\psi(q^*), \alpha, \boldsymbol{\gamma})$, where q^* is constructed using Algorithm 2. The proof of the following theorem can be obtained by making a few modifications to the proof of [5, Theorem 4.7], which are presented in [1].

Theorem 5.1 *Let b be prime, $s \geq 2$ and let $p \in \mathbb{Z}_b[x]$ be irreducible with $deg(p) = m \geq 1$. A minimizer q^* obtained from Algorithm 2 satisfies*

$$B(\psi(q^*), \alpha, \boldsymbol{\gamma}) \leq \frac{s^{1/\lambda}}{(b^m - 1)^{1/\lambda}} \prod_{j=1}^{s} \left(1 + \gamma_j^{\lambda} C_{b,\alpha,\lambda}\right)^{1/\lambda},$$

for all $\frac{1}{2\alpha} < \lambda \leq 1$, where $C_{b,\alpha,\lambda} > 0$ is given by (14).

We point out that the bounds in Theorems 4.1 and 5.1 only differ by the additional factor $s^{1/\lambda}$. We remark that the same observation was made in [5] and is also known from the lattice rule case. This leads to the conclusion that the Korobov construction is inferior to the component-by-component construction.

In the next corollary, we discuss the tractability of Algorithm 2.

Corollary 5.1 *Let b be prime, $s \geq 2$, $p \in \mathbb{Z}_b[x]$ be irreducible with $deg(p) = m \geq 1$ and $N = b^m$. Suppose $q^* \in R_{b,m}$ is constructed using Algorithm 2. Then we have the following:*

1.

$$B(\psi(q^*), \alpha, \boldsymbol{\gamma}) \leq c_{s,\alpha,\boldsymbol{\gamma},s}\, s^{2\alpha+1-\delta} (N-1)^{-(2\alpha+1)+\delta}, \text{ for all } 0 < \delta \leq 2\alpha,$$

where

$$c_{s,\alpha,\boldsymbol{\gamma},\delta} = \prod_{j=1}^{s} \left(1 + \gamma_j^{\frac{1}{2\alpha+1-\delta}} C_{b,\alpha,(2\alpha+1-\delta)^{-1}}\right)^{2\alpha+1-\delta}.$$

2. Under the assumption

$$A := \limsup_{s\to\infty} \frac{\sum_{j=1}^{s} \gamma_j}{\log s} < \infty$$

we obtain

$$c_{s,\alpha,\boldsymbol{\gamma},2\alpha} \leq \tilde{c}_\eta s^{\frac{A+\eta}{b^{2\alpha}-1}}$$

and therefore

$$B(\psi(q^*),\alpha,\boldsymbol{\gamma}) \leq \tilde{c}_\eta s^{1+\frac{A+\eta}{b^{2\alpha}-1}}(N-1)^{-1},$$

for all $\eta > 0$, where the constant $\tilde{c}_\eta$ only depends on η. Thus the bound $B(\psi(q^*),\alpha,\boldsymbol{\gamma})$ satisfies a bound which depends only polynomially on the dimension.

The proof is again similar to the proof of Corollary [5, Corollary 4.8] and can be found in [1].

6 A Lower Bound on the Worst-Case Variance

In this section, we produce a lower bound on the worst-case variance discussed in Section 3. As we rely on [20, Section 2.2.4, Proposition 1] to establish the result, the class of algorithms to which our result applies is the same as the class considered there. We now recall the definition of this class. Following [20, Section 1.1], we use the notation

$$S(f) = \int_{[0,1]^s} f(\boldsymbol{x})d\boldsymbol{x},$$

for $f \in V_{\alpha,s,\boldsymbol{\gamma}}$ and consider approximating $S : V_{\alpha,s,\boldsymbol{\gamma}} \to \mathbb{R}$ using a mapping $\tilde{S} : V_{\alpha,s,\boldsymbol{\gamma}} \to \mathbb{R}$. As in [20, Section 1.1], we assume that in general, the function $f \in V_{\alpha,s,\boldsymbol{\gamma}}$ is not known, but we have some information on f available, which is denoted by L, where $L : V_{\alpha,s,\boldsymbol{\gamma}} \to H$ and an approximation $\tilde{S} : V_{\alpha,s,\boldsymbol{\gamma}} \to \mathbb{R}$ only uses the information L if it can be written as follows $\tilde{S} = \varphi \circ L$, where $\varphi : H \to \mathbb{R}$ is a an arbitrary mapping, referred to as an (idealized) algorithm in [20]. In particular, we allow our approximation nodes to be chosen adaptively and define the following information operator:

$$\begin{aligned} I_N \;=\; & \big\{L : V_{\alpha,s,\boldsymbol{\gamma}} \to \mathbb{R}^N | L(f) = (f(\boldsymbol{a}_1), f(\boldsymbol{a}_2[f(\boldsymbol{a}_1)]), \ldots, f(\boldsymbol{a}_N[f(\boldsymbol{a}_1),\ldots,f(\boldsymbol{a}_{N-1})])) \,, \\ & \text{where } \boldsymbol{a}_1 \in [0,1]^s \text{ and } \boldsymbol{a}_i : \mathbb{R}^{i-1} \to [0,1]^s \text{ for } i = 2,\ldots,s\big\} \end{aligned}$$

and we can now introduce the class of all approximations considered in this section:

$$A_N = \left\{\tilde{S} : V_{\alpha,s,\boldsymbol{\gamma}} \to \mathbb{R} | \tilde{S} = \varphi \circ L \text{ with } \varphi : \mathbb{R}^N \to \mathbb{R} \text{ and } L \in I_N\right\}.$$

We remark that non-adaptive algorithms are of course included in A_N, consider $\tilde{S} = \varphi \circ \overline{L}$, where $\overline{L}(f) = (f(\boldsymbol{a}_1), \ldots f(\boldsymbol{a}_N))$. Now, following [20, Section 2.1], we can define the

randomized algorithms considered in this paper, referred to as generalized Monte Carlo methods in [20]: A random variable $Q = (Q(\omega))_{\omega\in\Omega}$ is called a randomized algorithm in A_N if (Ω, B, μ) is a probability space and $Q(\omega) \in A_N$ for all $\omega \in \Omega$. The set of all randomized algorithms is denoted by ${}^*C(A_N)$, hence randomly scrambled nets (and therefore polynomial lattice rules) are also included in this set. We now present the lower bound on the worst-case variance, which applies to all randomized algorithms in ${}^*C(A_N)$.

Theorem 6.1 *Let ${}^*C(A_N)$, $V_{\alpha,s,\boldsymbol{\gamma}}$ be defined as above. Then*

$$\inf_{Q\in{}^*C(A_N)} \sup_{f\in V_{\alpha,s,\boldsymbol{\gamma}}} \mathrm{Var}(\hat{I}(f,Q)) \geq \tilde{C} N^{-2\alpha-1},$$

for some constant $\tilde{C}$ independent of N where

$$\mathrm{Var}(\hat{I}(f,Q)) = \int_\Omega \left[\hat{I}(f,Q(\omega)) - \int_\Omega \hat{I}(f,Q(\omega'))\mathrm{d}\mu(\omega')\right]^2 \mathrm{d}\mu(\omega).$$

Proof. We remark that this proof follows along the lines of the proof of [8, Theorem 10]. We only consider $s = 1$, since integration in $V_{\alpha,1,\gamma_1}$ is no harder than integration in $V_{\alpha,s,\boldsymbol{\gamma}}$ with $s > 1$, as the one-dimensional space $V_{\alpha,1,\gamma_1}$ can be identified with the subspace of $V_{\alpha,s,\boldsymbol{\gamma}}$ consisting of functions depending only on the first variable. We let N be any given natural number and choose an integer m such that

$$b^{m-1} < 2N \leq b^m\,.$$

We define basic intervals

$$B_{m,a} = \left[\frac{a}{b^m}, \frac{a+1}{b^m}\right)\,,\ a = 0, 1, \ldots, b^m - 1\,,$$

and let $g_a(x) = \mathbf{1}_{B_{m,a}}(x)$ be the characteristic function of $B_{m,a}$. Then

$$\int_{[0,1]} g_a(x) g_c(x) dx = \begin{cases} b^{-m} & \text{if } a = c, \\ 0 & \text{otherwise.} \end{cases}$$

We now define

$$g = \sum_{a=0}^{b^m-1} \xi_a g_a\,,$$

where $\xi_a \in \{1,-1\}$ and bound $\sigma_l^2(g)$. Using Plancharel's identity we obtain that for any $l \geq 0$ we have

$$\sigma_l^2(g) \leq \sum_{l'=0}^{\infty} \sigma_{l'}^2(g) = \int_0^1 g^2(x)dx = \sum_{a,c=0}^{b^m-1} \xi_a \xi_c \int_0^1 g_a(x) g_c(x)dx = \frac{1}{b^m} \sum_{a=0}^{b^m-1} \xi_a^2 = 1.$$

Further, for $k \geq b^m$ we have

$$\widehat{g}(k) = \int_0^1 g(x)\overline{\mathrm{wal}_k(x)}dx = \sum_{a=0}^{b^m-1} \xi_a \int_0^1 g_a(x)\overline{\mathrm{wal}_k(x)}dx = \sum_{a=0}^{b^m-1} \xi_a \int_{a/b^m}^{(a+1)/b^m} \overline{\mathrm{wal}_k(x)}dx = 0,$$

since $\int_{a/b^m}^{(a+1)/b^m} \overline{\mathrm{wal}_k(x)}dx = 0$ for $k \geq b^m$ and hence for $l > m$ we have

$$\sigma_l^2(g) = \sum_{k=b^{l-1}}^{b^l-1} |\widehat{g}(k)|^2 = 0.$$

We set $f_a = \gamma_1 b^{-\alpha m} g_a$ for $a = 0, 1, \ldots, b^m - 1$. These f_a have disjoint support and

$$\int_{[0,1]} f_a(x)dx \geq \gamma_1 b^{-(\alpha+1)m}.$$

Set

$$f = \gamma_1 b^{-\alpha m} g = \sum_{a=0}^{b^m-1} \xi_a f_a,$$

then we get $\sigma_l^2(f) \leq \gamma_1^2 b^{-2\alpha m}$ for $0 \leq l \leq m$ and $\sigma_l^2(f) = 0$ for $l > m$. Hence

$$\|f\|_\alpha = \gamma_1^{-1} \sup_{l \in \mathbb{N}} b^{\alpha l} \sigma_l(f) \leq \gamma_1^{-1} \sup_{1 \leq l \leq m} b^{\alpha l} \gamma_1 b^{-\alpha m} \leq 1$$

and the result follows now from [20, Section 2.2.4, Proposition 1(ii)]. □

Remark 6.1 *For a large class of randomized algorithms, including adaptive ones, we have shown that the worst-case variance in the Walsh function space $V_{\alpha,s,\boldsymbol{\gamma}}$ behaves like $N^{-(1+2\alpha)}$. In Sections 4 and 5 we presented two algorithms which achieve worst-case variances of order $N^{-(1+2\alpha)+\delta}$, for all $\delta > 0$, and are hence almost optimal for the class of algorithms ${}^*C(A_N)$.*

7 Implementation of the Component-By-Component Algorithm

In this section, we show how to implement the CBC algorithm from Section 4. Our approach is based on [22], but we simplify the algorithm using ideas from [3]. Using ideas from [21, 22], we obtain, for $d \geq 2$,

$$\begin{aligned} & B(\boldsymbol{q}, \alpha, \boldsymbol{\gamma}) \\ &= \frac{1}{b^m} \sum_{h=0}^{b^m-1} \prod_{j=1}^{d} \left(1 + \frac{b}{b-1} \gamma_j \phi_\alpha(\boldsymbol{x}_{h,j})\right) - 1 \\ &= \frac{1}{b^m} \prod_{j=1}^{d} \left(1 + \frac{b}{b-1} \gamma_j \phi_\alpha(\boldsymbol{x}_{0,j})\right) - 1 + \frac{1}{b^m} \sum_{h=1}^{b^m-1} \boldsymbol{p}_{d-1}(h) \left(1 + \frac{b}{b-1} \gamma_d \phi_\alpha(\boldsymbol{x}_{h,d})\right), \end{aligned}$$

where

$$\boldsymbol{p}_{d-1}(h) = \prod_{j=1}^{d-1} \left(1 + \frac{b}{b-1} \gamma_j \phi_\alpha(\boldsymbol{x}_{h,j})\right).$$

Let $\omega\left(\frac{\overline{h}\overline{q}_d}{p}\right) = \phi_\alpha(\boldsymbol{x}_{h,d})$, where $\overline{h}$ and $\overline{q}_d$ denote the polynomials associated with h and q_d and p denotes the polynomial $p = p(x) \in \mathbb{Z}_b[x]$. Following [21], we now introduce the following matrix

$$\Omega_p = \left[\omega\left(\frac{\overline{h}\overline{q}}{p}\right)\right]_{\substack{q=1,\ldots,b^m-1 \\ h=1,\ldots,b^m-1}}, \tag{16}$$

i.e. rows are indexed by q and columns by h.

Let $\boldsymbol{p}_{d-1} = (\boldsymbol{p}_{d-1}(1), \ldots, \boldsymbol{p}_{d-1}(b^m-1))^\top$. Following [21], we have an update rule for $\boldsymbol{p}_d$ given by

$$\boldsymbol{p}_d = \operatorname{diag}\left(\left(\mathbf{1}_{(b^m-1)\times(b^m-1)} + \frac{b}{b-1}\gamma_d \Omega_p\right) v_{q_d}\right) \boldsymbol{p}_{d-1},$$

where $\operatorname{diag}(\boldsymbol{x})$ denotes the diagonal matrix with the elements of $\boldsymbol{x}$ on its diagonal and zero elsewhere and where we use v_j to denote a selection vector with 1 in position j and 0 elsewhere.

We now use the notation $B_{d-1} = (B((\boldsymbol{q}_{d-1}, \overline{1}), \alpha, \boldsymbol{\gamma}), \ldots, B((\boldsymbol{q}_{d-1}, \overline{b^m-1}), \alpha, \boldsymbol{\gamma}))^\top$. Then

$$B_{d-1} = \left[-1 + \frac{1}{b^m} \prod_{j=1}^{d} \left(1 + \frac{b}{b-1} \gamma_j \phi_\alpha(\boldsymbol{x}_{h,j})\right)\right] \mathbf{1}_{(b^m-1)\times 1}$$

$$
\begin{aligned}
&+\frac{1}{b^m}\left(\mathbf{1}_{(b^m-1)\times(b^m-1)}+\frac{b}{b-1}\gamma_d\Omega_p\right)\boldsymbol{p}_{d-1}\\
=\;&\left[-1+\frac{1}{b^m}\prod_{j=1}^{d}\left(1+\frac{b}{b-1}\gamma_j\phi_\alpha(\boldsymbol{x}_{0,j})\right)\right]\mathbf{1}_{(b^m-1)\times 1}\\
&+\frac{1}{b^m}\sum_{h=1}^{b^m-1}\boldsymbol{p}_{d-1}(h)\mathbf{1}_{(b^m-1)\times 1}+\frac{1}{b^m}\frac{b}{b-1}\gamma_d\Omega_p\boldsymbol{p}_{d-1}.
\end{aligned}
$$

In the next lemma, we summarize an observation from [3]. Let

$$\Pi(g)=[\Pi_{k,l}]_{\substack{k=1,\ldots,b^m-1\\ l=1,\ldots,b^m-1}}$$

where

$$\Pi_{k,l}=\begin{cases}1 & \text{if } \overline{k}(x)\equiv g^l(x) \pmod p\\ 0 & \text{otherwise}\end{cases} \tag{17}$$

and

$$\Pi(g^{-1})=\left[\Pi_{k,l}^{-1}\right]_{\substack{k=1,\ldots,b^m-1\\ l=1,\ldots,b^m-1}}$$

where

$$\Pi_{k,l}^{-1}=\begin{cases}1 & \text{if } \overline{k}(x)\equiv g^{-l}(x) \pmod p\\ 0 & \text{otherwise,}\end{cases} \tag{18}$$

be two permutation matrices, where g is a primitive element which generates all elements of $(\mathbb{Z}_b[x]/p)^*=\left\{g^0,g^1,\ldots,g^{b^m-1}\right\}$; such an element g is known to exist since the multiplicative group of every finite field is cyclic. Let $t_k=\deg(g^k \pmod p)$, $k=0,1,\ldots,b^m-2$, and set

$$A_3=\left[b^{2\alpha t_{i-j} \pmod{b^m-1}}\right]_{\substack{i=1,\ldots,b^m-1\\ j=1,\ldots,b^m-1}} \tag{19}$$

and note that A_3 is a circulant matrix, which allows us to use Fast Fourier Transforms (FFTs) as in [21, 22]. We now state the lemma.

Lemma 7.1 *Let p be an irreducible polynomial, let g be a primitive element of $(\mathbb{Z}_b[x]/p)^*$, and let $\Pi(g)$, $\Pi(g^{-1})$, A_3 and Ω_p be defined as above. Then*

$$\Omega_p=\mathbf{1}_{(b^m-1)\times(b^m-1)}\frac{b-1}{b(b^{2\alpha}-1)}-\frac{b^{2\alpha+1}-1}{b(b^{2\alpha}-1)}b^{-2\alpha m}\Pi(g)A_3\Pi(g^{-1})^\top.$$

Proof. It follows from the definition of $\phi_\alpha(x)$

$$\phi_\alpha\left(v_m\left(\frac{\overline{h}\overline{q}}{p}\right)\right) = \frac{b-1}{b(b^{2\alpha}-1)} - \frac{(b^{2\alpha+1}-1)b^{-2\alpha a_{0,h,q}}}{b(b^{2\alpha}-1)}\,,$$

where $a_{0,h,q}$ denotes the smallest integer a so that $\xi_{h,q,a} \neq 0$, and where

$$v_m\left(\frac{\overline{h}\overline{q}}{q}\right) = \frac{\xi_{h,q,1}}{b} + \frac{\xi_{h,q,2}}{b^2} + \dots.$$

Hence

$$\Omega_p = \frac{b-1}{b(b^{2\alpha}-1)}\mathbf{1}_{(b^m-1)\times(b^m-1)} - \frac{b^{2\alpha+1}-1}{b(b^{2\alpha}-1)}A_1,$$

where

$$A_1 = \left[b^{-2\alpha a_{0,h,q}}\right]_{\substack{q=1,\dots,b^m-1\\ h=1,\dots,b^m-1}}.$$

Now assume that for $w \in \mathbb{Z}_b[x]$ we have

$$\frac{w(x)}{p(x)} = u_{1,w}x^{-1} + u_{2,w}x^{-2} + \dots, \tag{20}$$

where $u_{j,w} \in \mathbb{Z}_b$. Then

$$v_m\left(\frac{\overline{h}\overline{q}}{p}\right) = u_{1,\overline{h}\overline{q}}b^{-1} + u_{2,\overline{h}\overline{q}}b^{-2} + \dots + u_{m,\overline{h}\overline{q}}b^{-m}\,,$$

hence, $a_{0,h,q}$ is the smallest integer a so that $u_{a,\overline{h}\overline{q}} \neq 0$, $\overline{h}, \overline{q} \in \mathbb{Z}_b[x]$ (note that $p \not| \overline{h}, \overline{q}$).

The matrix A_2 given by

$$A_2 = \Pi^\top(g)A_1\Pi(g^{-1})$$

is circulant. Indeed it can be checked that

$$A_2 = \left[b^{-2\alpha a_{0,g^{-j},g^i}}\right]_{\substack{i=1,\dots,b^m-1\\ j=1,\dots,b^m-1}}, \tag{21}$$

where g^{-j} and g^i in Equation (21) denote the integers associated with the polynomials $g^{-j} \pmod p$ and $g^i \pmod p$. We let $a_{0,g^{-j},g^i} = r_{i-j}$ and note that $r_k = r_{k'}$ for $k \equiv k' \pmod{b^m-1}$, as $g^{b^m-1} \equiv 1 \pmod p$, hence

$$A_2 = \left[b^{-2\alpha r_{i-j}}\right]_{\substack{i=1,\dots,b^m-1\\ j=1,\dots,b^m-1}}.$$

The matrix A_2 is circulant and r_k is the smallest integer r such that $u_{r,g^k} \neq 0$, which implies using Equation (20) that

$$\deg(g^k \pmod p) = \deg(p) - r_k \,,$$

and consequently

$$r_k = m - \deg(g^k \pmod p) \,.$$

Now denoting

$$t_k = \deg(g^k \pmod p) \,,$$

we get

$$A_2 = b^{-2\alpha m} A_3 \,,$$

where A_3 is given by Equation (19) and the result follows. □

Note that if the polynomial p in the lemma above is primitive, then one can choose the primitive element $g(x) = x$. In Algorithm 3 we show how to implement the CBC algorithm from Section 4. Several remarks regarding Algorithm 3 are in order.

Algorithm 3 Fast CBC algorithm

Require: b a prime, $s, m \in \mathbb{N}$ and weights $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$.

1: Choose a primitive polynomial $p \in \mathbb{Z}_b[x]$, with $\deg(p) = m$, and choose $g(x) = x$.
2: $\boldsymbol{\mu}_0 := \mathbf{1}_{(b^m - 1 \times 1)}$.
3: **for** $d = 1$ to s **do**
4: $\quad \tilde{B}_d = A_3 \boldsymbol{\mu}_{d-1}$.
5: $\quad w_d = \arg\min_{w \in R_{b,m}} \tilde{B}_d(w)$.
6:

$$\boldsymbol{\mu}_d = \operatorname{diag}\left(\mathbf{1}_{1\times(b^m-1)}\left(1 + \frac{\gamma_d}{(b^{2\alpha}-1)}\right) - \gamma_d \frac{(b^{2\alpha+1}-1)b^{-2\alpha m}}{(b-1)(b^{2\alpha}-1)} A_3(w_d, :)\right)\boldsymbol{\mu}_{d-1} \,.$$

7: **end for**
8: **return** $\boldsymbol{q} = (q_1, \ldots, q_s)$.

Remark 7.1 *As in [21, 22], we search for the minimum in the permuted space, hence we minimize* $\tilde{B}_d = \Pi^\top(g) B_{d-1}$. *However, as in [21, 22], the component* z_d *can be found by mapping back* w_d *using* $\Pi(g)$.

Remark 7.2 *We have* $\boldsymbol{\mu}_d = \Pi^\top(g^{-1})\boldsymbol{p}_d$ *and consequently update* $\boldsymbol{\mu}_d$ *using* $\boldsymbol{\mu}_{d-1}$*. Hence we do not need to permute back and forth, but can complete the algorithm in the permuted space.*

The next corollary gives information on the computational complexity of Algorithm 3. We use

$$\boldsymbol{a} = \begin{bmatrix} b^{2\alpha t_0} \\ b^{2\alpha t_1} \\ \vdots \\ b^{2\alpha t_{b^m-2}}. \end{bmatrix} \tag{22}$$

to denote the vector generating the circulant matrix A_3 in Lemma 7.1 and Algorithm 3.

Corollary 7.1 *Assume that the vector* $\boldsymbol{a}$ *in Equation* (22) *has been precomputed and stored using* $\mathcal{O}(b^m)$ *memory. Then Algorithm 3 can be completed in time* $\mathcal{O}(sb^m m)$ *and memory* $\mathcal{O}(b^m)$*.*

For a proof, see [21, 22] or also [7, Section 10.3].

8 Numerical Experiments

In this section, we numerically investigate the performance of the CBC algorithm presented in Section 4; we rely on Section 7 for the implementation of the algorithm. In Tables 1 - 3, we present values of $B(\boldsymbol{q}, \alpha, \boldsymbol{\gamma})$ for different choices of α and $\boldsymbol{\gamma}$, where $\boldsymbol{q}$ is constructed using Algorithm 3.

We compare the performance of the CBC algorithm to the performance of digital nets. As was done with scrambled polynomial lattice rules in Section 3, we can study the variance of the estimator $\hat{I}(f)$ given in Equation (10), consider the worst-case variance of multivariate integration in $V_{\alpha,s,\boldsymbol{\gamma}}$ and bound this variance as follows:

$$\mathrm{Var}(Q_{b^m,s}(C_1,\ldots,C_s), V_{\alpha,s,\boldsymbol{\gamma}}) \leq \sum_{\boldsymbol{k}\in\mathscr{D}(C_1,\ldots,C_s)\setminus\{\boldsymbol{0}\}} r_{2\alpha+1,\boldsymbol{\gamma}}(\boldsymbol{k}), \tag{23}$$

where $C_1,\ldots,C_s$ are the generating matrices of the digital net under consideration and $\mathscr{D}(C_1,\ldots,C_s)$ is its dual space. We denote the bound (23) by $B((C_1,\ldots,C_s),\alpha,\boldsymbol{\gamma})$, and remark that $B((C_1,\ldots,C_s),\alpha,\boldsymbol{\gamma})$ can also be computed using Equation (12), where $\{\boldsymbol{x}_h\}_{h=0}^{b^m-1}$ is the digital net generated by $C_1,\ldots,C_s$.

Consequently, we compare the values of $B(\boldsymbol{q},\alpha,\boldsymbol{\gamma})$ to the values of $B((C_1,\ldots,C_s),\alpha,\boldsymbol{\gamma})$ in Tables 1 - 3; in each cell, the top number corresponds to the CBC construction and

the bottom one to the digital net. We choose the following digital nets: For $s = 1$, we simply choose equidistributed points, $x_h = \frac{h}{b^m}$, $h = 0, \ldots, b^m - 1$, for $s = 5$, we use Pirsic's implementation of Niederreiter-Xing points, [26], and for $s = 50$ and $s = 100$, we use Sobol points as constructed in [11]; we point out that for the CBC construction, we choose $b = 2$ and likewise, the digital nets under consideration are digital nets over $\mathbb{Z}_2$.

We derive the following conclusions from the tables: For $s = 1$, as expected, we obtain the optimal rate of convergence, $2^{-(2\alpha+1)m}$, and observe the same values for the CBC construction as for the digital nets. Regarding the case $s = 5$, the values are comparable, however, the Niederreiter-Xing construction seems to be slightly better than the CBC construction for the examples considered. Finally, for $s = 50$ and $s = 100$, the performances of the two methods are again comparable, however, this time, the CBC construction seems to outperform the digital nets.

References

[1] J. Baldeaux, Higher order nets and sequences, PhD thesis, The University of New South Wales, 2010.

[2] R. E. Caflish, W. J. Morokoff, A. B. Owen, Valuation of mortgage backed securities using Brownian Bridges to reduce effective dimension, J. Comput. Finance, 1, 27–46, 1997.

[3] J. Dick, On the fast component-by-component algorithm for polynomial lattice rules, Available at: http://quasirandomideas.wordpress.com/2009/12/31/fast-cbc-for-polynomial-lattice-rules, Posted on December 31st, 2009, Last accessed February 2nd, 2010.

[4] J. Dick, M. Gnewuch, Embedding Theorems for Fractional Spaces and Numerical Integration, In preparation.

[5] J. Dick, F. Kuo, F. Pillichshammer, I. Sloan, Construction algorithms for polynomial lattice rules for multivariate integration, Math. Comp., 74, 1895–1921, 2005.

[6] J. Dick, F. Pillichshammer, Multivariate integration in weighted Hilbert spaces based on Walsh functions and weighted Sobolev spaces, J. Complexity, 21, 149–195, 2005.

[7] J. Dick, F. Pillichshammer, Digital Nets and Sequences, Discrepancy and Quasi-Monte Carlo Integration, Cambridge University Press, Cambridge, 2010 (to appear).

	$\alpha = 0.5$				$\alpha = 1$			
$m =$	$s = 1$	$s = 5$	$s = 50$	$s = 100$	$s = 1$	$s = 5$	$s = 50$	$s = 100$
4	3.91e-03	1.46e+00	7.04e+13	7.92e+28	8.14e-05	4.37e-02	1.10e+05	1.95e+11
	3.91e-03	1.48e+00	7.04e+13	7.92e+28	8.14e-05	4.90e-02	1.10e+05	1.95e+11
5	9.77e-04	6.16e-01	3.52e+13	3.96e+28	1.02e-05	1.09e-02	5.52e+04	9.74e+10
	9.77e-04	6.34e-01	3.52e+13	3.96e+28	1.02e-05	1.32e-02	5.52e+04	9.74e+10
6	2.44e-04	2.66e-01	1.76e+13	1.98e+28	1.27e-06	3.45e-03	2.76e+04	4.87e+10
	2.44e-04	2.61e-01	1.76e+13	1.98e+28	1.27e-06	3.17e-03	2.76e+04	4.87e+10
7	6.10e-05	1.08e-01	8.80e+12	9.90e+27	1.59e-07	9.05e-04	1.38e+04	2.44e+10
	6.10e-05	1.04e-01	8.80e+12	9.90e+27	1.59e-07	7.19e-04	1.38e+04	2.44e+10
8	1.53e-05	4.24e-02	4.40e+12	4.95e+27	1.99e-08	2.36e-04	6.90e+03	1.22e+10
	1.53e-05	3.93e-02	4.40e+12	4.95e+27	1.99e-08	1.48e-04	6.90e+03	1.22e+10
9	3.81e-06	1.74e-02	2.20e+12	2.48e+27	2.48e-09	6.10e-05	3.45e+03	6.09e+09
	3.81e-06	1.44e-02	2.20e+12	2.48e+27	2.48e-09	2.86e-05	3.45e+03	6.09e+09
10	9.54e-07	6.41e-03	1.10e+12	1.24e+27	3.10e-10	1.29e-05	1.72e+03	3.04e+09
	9.54e-07	5.21e-03	1.10e+12	1.24e+27	3.10e-10	5.56e-06	1.72e+03	3.04e+09
11	2.38e-07	2.29e-03	5.50e+11	6.19e+26	3.88e-11	2.56e-06	8.62e+02	1.52e+09
	2.38e-07	1.82e-03	5.50e+11	6.19e+26	3.88e-11	1.01e-06	8.62e+02	1.52e+09
12	5.96e-08	8.39e-04	2.75e+11	3.09e+26	4.85e-12	5.03e-07	4.31e+02	7.61e+08
	5.96e-08	6.17e-04	2.75e+11	3.09e+26	4.85e-12	1.78e-07	4.31e+02	7.61e+08
13	1.49e-08	3.09e-04	1.37e+11	1.55e+26	6.06e-13	1.05e-07	2.15e+02	3.81e+08
	1.49e-08	2.06e-04	1.37e+11	1.55e+26	6.06e-13	3.07e-08	2.16e+02	3.81e+08
14	3.73e-09	1.12e-04	6.87e+10	7.74e+25	7.58e-14	2.56e-08	1.08e+02	1.90e+08
	3.73e-09	6.76e-05	6.87e+10	7.74e+25	7.57e-14	5.17e-09	1.08e+02	1.90e+08
15	9.31e-10	3.66e-05	3.44e+10	3.87e+25	9.27e-15	4.98e-09	5.38e+01	9.52e+07
	9.31e-10	2.18e-05	3.44e+10	3.87e+25	9.33e-15	8.54e-10	5.39e+01	9.52e+07
16	2.33e-10	1.29e-05	1.72e+10	1.93e+25	1.22e-15	8.92e-10	2.69e+01	4.76e+07
	2.33e-10	6.94e-06	1.72e+10	1.93e+25	1.11e-15	1.38e-10	2.69e+01	4.76e+07

Table 1: Values of $B(\boldsymbol{q}, \alpha, \boldsymbol{\gamma})$ and $B((C_1, \ldots, C_s), \alpha, \boldsymbol{\gamma})$ for $\gamma_j = 1$, $j = 1, \ldots, s$ and $\boldsymbol{q}$ constructed using the CBC algorithm; the top number gives the value of $B(\boldsymbol{q}, \alpha, \boldsymbol{\gamma})$, the bottom the value of $B((C_1, \ldots, C_s), \alpha, \boldsymbol{\gamma})$.

	$\alpha = 0.5$				$\alpha = 1$			
$m =$	$s = 1$	$s = 5$	$s = 50$	$s = 100$	$s = 1$	$s = 5$	$s = 50$	$s = 100$
4	3.42e-03	4.64e-01	2.03e+01	2.04e+01	7.12e-05	1.47e-02	2.82e-01	2.83e-01
	3.42e-03	4.84e-01	2.04e+01	2.06e+01	7.12e-05	1.83e-02	3.46e-01	3.48e-01
5	8.54e-04	1.87e-01	9.99e+00	1.01e+01	8.90e-06	3.54e-03	1.16e-01	1.17e-01
	8.54e-04	1.95e-01	1.01e+01	1.01e+01	8.90e-06	4.45e-03	1.38e-01	1.39e-01
6	2.14e-04	7.75e-02	4.91e+00	4.95e+00	1.11e-06	1.04e-03	4.78e-02	4.82e-02
	2.14e-04	7.46e-02	4.94e+00	4.98e+00	1.11e-06	9.29e-04	5.30e-02	5.34e-02
7	5.34e-05	2.96e-02	2.40e+00	2.42e+00	1.39e-07	2.54e-04	1.85e-02	1.87e-02
	5.34e-05	2.80e-02	2.44e+00	2.47e+00	1.39e-07	1.97e-04	2.39e-02	2.41e-02
8	1.34e-05	1.17e-02	1.17e+00	1.18e+00	1.74e-08	5.77e-05	7.39e-03	7.45e-03
	1.34e-05	1.01e-02	1.20e+00	1.21e+00	1.74e-08	3.79e-05	1.08e-02	1.09e-02
9	3.34e-06	4.43e-03	5.66e-01	5.71e-01	2.17e-09	1.29e-05	2.84e-03	2.87e-03
	3.34e-06	3.54e-03	5.89e-01	5.95e-01	2.17e-09	6.98e-06	4.94e-03	4.97e-03
10	8.34e-07	1.56e-03	2.72e-01	2.75e-01	2.72e-10	3.03e-06	1.08e-03	1.09e-03
	8.34e-07	1.22e-03	2.88e-01	2.90e-01	2.72e-10	1.28e-06	2.24e-03	2.26e-03
11	2.09e-07	5.45e-04	1.30e-01	1.31e-01	3.40e-11	6.24e-07	4.01e-04	4.06e-04
	2.09e-07	4.10e-04	1.42e-01	1.43e-01	3.40e-11	2.22e-07	9.58e-04	9.66e-04
12	5.22e-08	1.93e-04	6.20e-02	6.26e-02	4.24e-12	1.16e-07	1.49e-04	1.51e-04
	5.22e-08	1.35e-04	6.73e-02	6.80e-02	4.24e-12	3.79e-08	3.59e-04	3.64e-04
13	1.30e-08	7.07e-05	2.94e-02	2.97e-02	5.31e-13	2.48e-08	5.43e-05	5.51e-05
	1.30e-08	4.38e-05	3.33e-02	3.36e-02	5.30e-13	6.34e-09	2.11e-04	2.13e-04
14	3.26e-09	2.27e-05	1.39e-02	1.40e-02	6.62e-14	4.56e-09	1.99e-05	2.02e-05
	3.26e-09	1.40e-05	1.58e-02	1.60e-02	6.62e-14	1.04e-09	8.23e-05	8.31e-05
15	8.15e-10	8.01e-06	6.49e-03	6.57e-03	8.49e-15	1.10e-09	7.15e-06	7.26e-06
	8.15e-10	4.41e-06	7.70e-03	7.78e-03	8.22e-15	1.67e-10	4.44e-05	4.48e-05
16	2.04e-10	2.68e-06	3.02e-03	3.06e-03	9.99e-16	1.81e-10	2.57e-06	2.61e-06
	2.04e-10	1.37e-06	3.76e-03	3.80e-03	8.88e-16	2.64e-11	2.14e-05	2.15e-05

Table 2: Values of $B(\boldsymbol{q}, \alpha, \boldsymbol{\gamma})$ and $B((C_1, \ldots, C_s), \alpha, \boldsymbol{\gamma})$ for $\gamma_j = 0.875^j$, $j = 1, \ldots, s$ and $\boldsymbol{q}$ constructed using the CBC algorithm; the top number gives the value of $B(\boldsymbol{q}, \alpha, \boldsymbol{\gamma})$, the bottom the value of $B((C_1, \ldots, C_s), \alpha, \boldsymbol{\gamma})$.

	$\alpha = 0.5$				$\alpha = 1$			
$m =$	$s = 1$	$s = 5$	$s = 50$	$s = 100$	$s = 1$	$s = 5$	$s = 50$	$s = 100$
4	3.91e-03	2.75e-02	4.75e-02	4.90e-02	8.14e-05	7.68e-04	1.80e-03	1.88e-03
	3.91e-03	3.20e-02	5.97e-02	6.17e-02	8.14e-05	1.27e-03	4.40e-03	4.62e-03
5	9.77e-04	8.98e-03	1.78e-02	1.84e-02	1.02e-05	1.48e-04	4.88e-04	5.20e-04
	9.77e-04	1.25e-02	2.22e-02	2.30e-02	1.02e-05	3.13e-04	1.23e-03	1.30e-03
6	2.44e-04	2.95e-03	6.29e-03	6.56e-03	1.27e-06	3.37e-05	1.20e-04	1.31e-04
	2.44e-04	3.20e-03	7.23e-03	7.66e-03	1.27e-06	3.62e-05	2.47e-04	2.89e-04
7	6.10e-05	8.96e-04	2.22e-03	2.35e-03	1.59e-07	5.05e-06	2.91e-05	3.31e-05
	6.10e-05	1.11e-03	2.65e-03	2.88e-03	1.59e-07	8.38e-06	6.91e-05	9.12e-05
8	1.53e-05	2.96e-04	7.86e-04	8.36e-04	1.99e-08	1.04e-06	6.94e-06	8.16e-06
	1.53e-05	3.44e-04	1.00e-03	1.12e-03	1.99e-08	1.37e-06	2.16e-05	3.52e-05
9	3.81e-06	9.34e-05	2.81e-04	3.02e-04	2.48e-09	1.90e-07	1.67e-06	2.02e-06
	3.81e-06	9.15e-05	3.27e-04	3.64e-04	2.48e-09	1.60e-07	4.70e-06	6.18e-06
10	9.54e-07	2.78e-05	9.54e-05	1.03e-04	3.10e-10	4.07e-08	4.20e-07	5.14e-07
	9.54e-07	2.69e-05	1.19e-04	1.32e-04	3.10e-10	2.49e-08	1.83e-06	2.37e-06
11	2.38e-07	8.95e-06	3.34e-05	3.64e-05	3.88e-11	6.34e-09	9.59e-08	1.21e-07
	2.38e-07	8.25e-06	4.10e-05	4.71e-05	3.88e-11	3.76e-09	3.25e-07	5.44e-07
12	5.96e-08	2.68e-06	1.18e-05	1.30e-05	4.85e-12	1.30e-09	2.36e-08	3.03e-08
	5.96e-08	2.54e-06	1.46e-05	1.68e-05	4.85e-12	6.34e-10	1.16e-07	1.64e-07
13	1.49e-08	8.29e-07	4.05e-06	4.50e-06	6.06e-13	2.04e-10	5.79e-09	7.61e-09
	1.49e-08	7.08e-07	4.69e-06	5.76e-06	6.06e-13	9.21e-11	2.45e-08	5.16e-08
14	3.73e-09	2.50e-07	1.40e-06	1.57e-06	7.58e-14	4.11e-11	1.40e-09	1.88e-09
	3.73e-09	1.97e-07	1.60e-06	1.98e-06	7.57e-14	1.29e-11	7.04e-09	2.04e-08
15	9.31e-10	7.70e-08	4.91e-07	5.54e-07	9.27e-15	6.15e-12	3.45e-10	4.79e-10
	9.31e-10	5.59e-08	5.50e-07	7.09e-07	9.33e-15	1.74e-12	2.52e-09	4.52e-09
16	2.33e-10	2.34e-08	1.71e-07	1.95e-07	1.22e-15	1.00e-12	8.51e-11	1.22e-10
	2.33e-10	1.69e-08	1.89e-07	2.52e-07	1.11e-15	2.70e-13	9.54e-10	1.45e-09

Table 3: Values of $B(\boldsymbol{q}, \alpha, \boldsymbol{\gamma})$ and $B((C_1, \ldots, C_s), \alpha, \boldsymbol{\gamma})$ for $\gamma_j = j^{-2}$, $j = 1, \ldots, s$ and $\boldsymbol{q}$ constructed using the CBC algorithm; the top number gives the value of $B(\boldsymbol{q}, \alpha, \boldsymbol{\gamma})$, the bottom the value of $B((C_1, \ldots, C_s), \alpha, \boldsymbol{\gamma})$.

[8] S. Heinrich, F. J. Hickernell, R. X. Yue, Optimal quadrature for haar wavelet spaces, Math. Comp., 73, 259–277, 2004.

[9] F.J. Hickernell, H. Woźniakowski, The Price of Pessimism for Multidimensional Quadrature, J. Complexity, 17, 625–659, 2001.

[10] F.J. Hickernell, R.X. Yue, The mean square discrepancy of scrambled (t, s)-sequences, SIAM J. Numer. Anal., 38, 1089–1112, 2000.

[11] S. Joe, F.Y. Kuo, Remark on Algorithm 659: Implementing Sobol's quasirandom sequence generator, ACM Trans. Math. Softw., 29, 49–57, 2003.

[12] N.M. Korobov, Properties and calculation of optimal coefficients, Dokl. Akad. Nauk SSSR, 132, 1009-1012, 1960 (in Russian).

[13] G. Larcher, A. Lauss, H. Niederreiter, W. Ch. Schmid, Optimal polynomials for (t, m, s)-nets and numerical integration of multivariate Walsh series, SIAM J. Numer. Anal., 33, 2239 – 2253, 1996.

[14] G. Larcher, C. Traunfellner, On the numerical integration of Walsh series by number-theoretic methods, Math Comp., 63, 277–291, 1994.

[15] Ch. Lemieux, P. L'Ecuyer, Randomized Polynomial Lattice Rules for Multivariate Integration and Simulation, SIAM J. Sci. Comput., 24, 1768–1789, 2003.

[16] J. Matoušek, Geometric Discrepancy, Algorithms and Combinatorics 18, Springer Verlag, Berlin, 1999.

[17] H. Niederreiter, Point sets and sequences with small discrepancy, Monatsh. Math., 104, 273–337, 1987.

[18] H. Niederreiter, Random number generation and quasi-Monte Carlo methods, CBMS-NSF Regional Conference Series in Applied Mathematics, Vol. 63, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[19] H. Niederreiter, Low-discrepancy point sets obtained by digital constructions over finite fields, Czech. Math. J., 42, 143–166, 1992.

[20] E. Novak, Deterministic and stochastic error bounds in numerical analysis, Lectures in Notes in Math., no. 1349, Springer-Verlag, Berlin, 1988.

[21] D. Nuyens, R. Cools, Fast algorithms for component-by-component constructions of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces, Math. Comp., 75, 903–920, 2006.

[22] D. Nuyens, R. Cools, Fast component-by-component construction, a reprise for different kernels, In: Monte Carlo and Quasi-Monte Carlo Methods 2004, H. Niederreiter and D. Talay (eds.), 373 – 388, Springer, Berlin, 2006.

[23] A.B. Owen, Randomly permuted (t, m, s)-nets and (t, s)-sequences, In: Monte Carlo and quasi-Monte Carlo Methods in Scientific Computing, H. Niederreiter and P. Jau-Shyong Shiue (eds.), 299–317, Springer, New York, 1995.

[24] A. B. Owen, Monte Carlo variance of scrambled net quadrature, SIAM J. Numer. Anal., 34, 1884–1910, 1997.

[25] A.B. Owen, Scrambled net variance for integrals of smooth functions, Ann. Statist., 25, 1541–1562, 1997.

[26] G. Pirsic, A software implementation of Niederreiter-Xing sequences, In: Monte Carlo and quasi-Monte Carlo methods 2000, K.T. Fang, F.J. Hickernell, and H. Niederreiter (eds.), 434–445, Springer Verlag, Berlin, 2002.

[27] I. H. Sloan, S. Joe, Lattice methods for multiple integration, Oxford Science Publications, The Clarendon Press Oxford University Press, New York, 1994.

[28] I.H. Sloan, A.V. Reztsov, Component-by-component construction of good lattice rules, Math. Comp., 71, 263–273, 2002.

[29] I. H. Sloan, H. Woźniakowski, When are quasi-Monte Carlo algorithms efficient for high dimensional integrals?, J. Complexity, 14, 1–33, 1998.

[30] R.X. Yue, Variance of quadrature over scrambled unions of nets, Statist. Sinica, 9, 451–473, 1999.

[31] R.X. Yue, F.J. Hickernell, The discrepancy and gain coefficients of scrambled digital nets, J. Complexity, 18, 135–151, 2002.

[32] R.X. Yue, F.J. Hickernell, Strong tractability of integration using scrambled Niederreiter points, Math. Comp., 74, 1871–1893, 2005.

[33] R.X. Yue, S.S Mao, On the variance of quadrature over scrambled nets and sequences, Statist. Probab. Lett., 44, 267–280, 1999.

Author's Addresses:

Jan Baldeaux, School of Mathematics and Statistics, The University of New South Wales, Sydney 2052, Australia. Email: Jan.Baldeaux@unsw.edu.au

Josef Dick, School of Mathematics and Statistics, The University of New South Wales, Sydney 2052, Australia. Email: josef.dick@unsw.edu.au